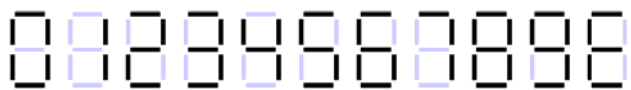
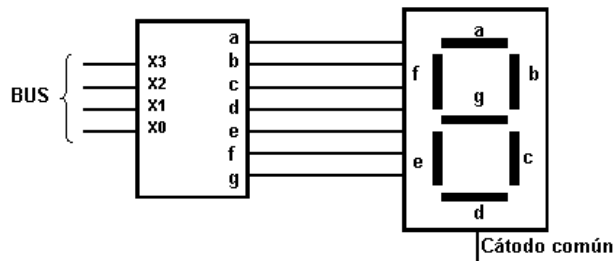


ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES I. FINAL 12-6-05
SOLUCIONES

P1.- Un display 7-segmentos es un dispositivo electrónico compuesto por 7 diodos LED, emisores de luz, dispuestos y nombrados según se muestra en la figura de la derecha. Encendiendo dichos diodos o segmentos de la manera correcta se pueden representar los números del 0 al 9 de una forma sencilla, según se muestra en la figura inferior.



Se desea visualizar el número decimal contenido en un bus de 4 bits, para lo que se va a utilizar un display de 7-segmentos. Para ello, se necesita un circuito que decodifique el contenido del bus de 4 bits a la información correspondiente en 7-segmentos, es decir, un decodificador BCD a 7-segmentos. En el caso de que el dato contenido en el bus no se corresponda con un número decimal, en el display se presentará el símbolo de error (una E mayúscula).

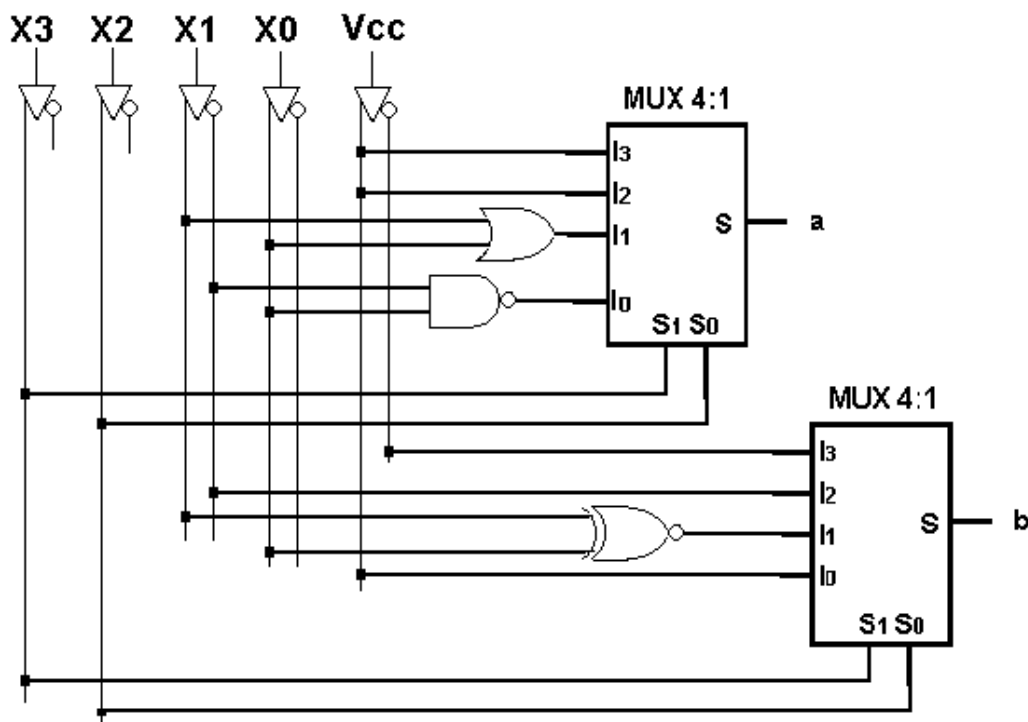


Se pide:

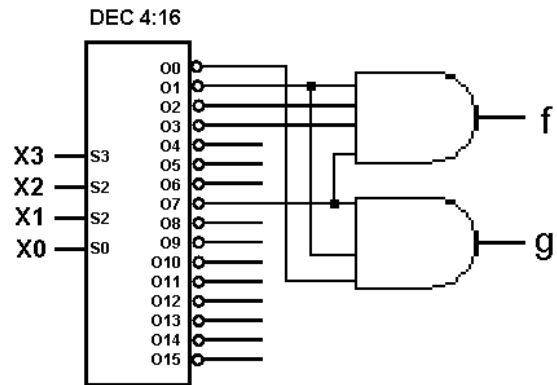
a) Rellenar la tabla de verdad del decodificador pedido.

BUS	X3	X2	X1	X0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	0	0	1	1	1	1
B	1	0	1	1	1	0	0	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	1
D	1	1	0	1	1	0	0	1	1	1	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	1	1	1	1

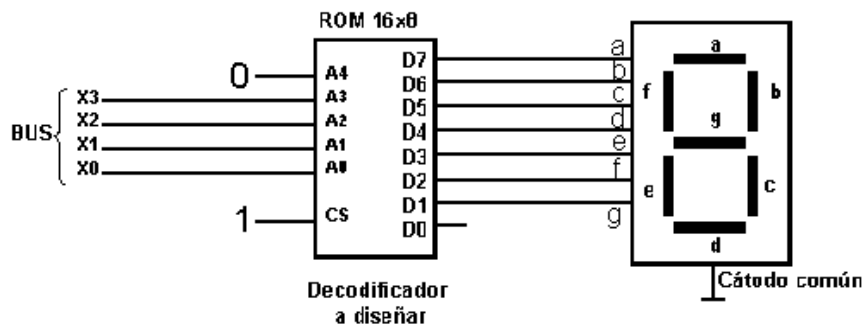
b) Implementar las salidas a y b utilizando un multiplexor 4:1 para cada una de ellas y el mínimo número de puertas lógicas posible.



- c) Implementar las salidas f y g utilizando un decodificador 4:16 con las salidas negadas y el mínimo número de puertas lógicas posible.



- d) Implementar todas las funciones utilizando una memoria de 32x8. Indicar las conexiones de la memoria con el bus y el display en la figura adjunta, así como el programa contenido en la memoria, relleno la tabla.

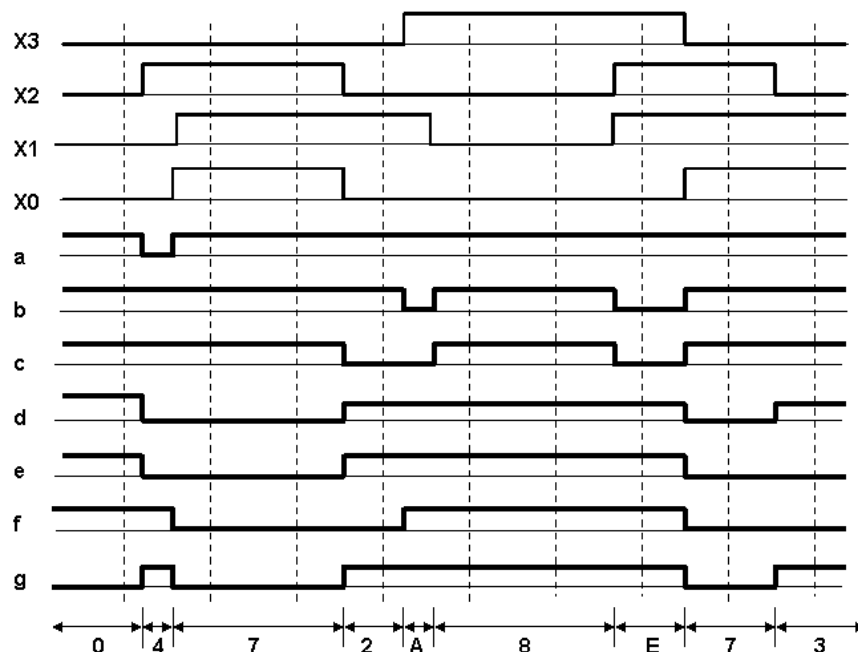


Se muestra una de las múltiples soluciones posibles respecto a la conexión entre la memoria y el display (se podrían haber usado las salidas desde D0 a D6, haber conectado los segmentos con otro orden, etc) y la asignación de bits de las entradas de

dirección.

Direcc.	Dato	Direcc.	Dato	Direcc.	Dato	Direcc.	Dato
00	FC	08	FE	10	XX	18	XX
01	60	09	F6	11	XX	19	XX
02	DA	0A	9E	12	XX	1A	XX
03	F2	0B	9E	13	XX	1B	XX
04	66	0C	9E	14	XX	1C	XX
05	B6	0D	9E	15	XX	1D	XX
06	BE	0E	9E	16	XX	1E	XX
07	F0	0F	9E	17	XX	1F	XX

- e) Completar el cronograma siguiente para las señales de entrada mostradas. Suponer que no hay retardo en el decodificador.



P2.- Dada la siguiente tabla de estados:

Estado actual $Q_1^n Q_0^n$	Estado siguiente/salida ($Q_1^{n+1} Q_0^{n+1}/Z$)			
	$X_1 X_0 = 00$	$X_1 X_0 = 01$	$X_1 X_0 = 10$	$X_1 X_0 = 11$
00	01/0	00/0	01/0	00/0
01	10/0	00/0	10/0	00/0
10	10/0	11/0	10/0	11/0
11	01/0	00/1	00/1	00/0

a) Indicar si la máquina a la que representa es Moore o Mealy, justificando la respuesta

Es una máquina Mealy puesto que la salida depende de las entradas del sistema

b) Calcular las ecuaciones de excitación y la ecuación de la salida

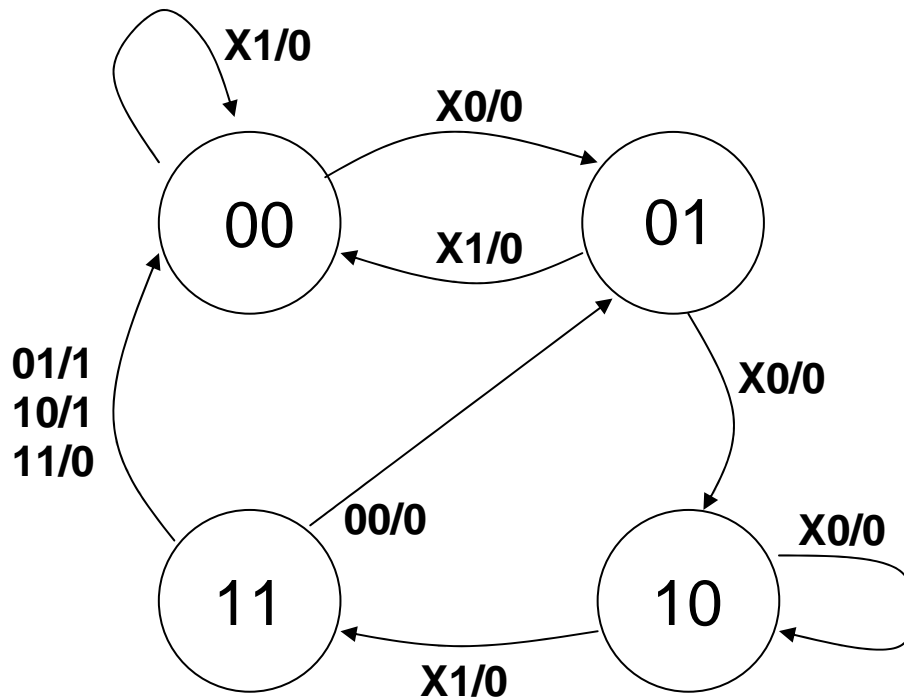
$$J_1 = \neg X_0 Q_0$$

$$K_1 = Q_0 \quad J_0 = \neg X_0 / Q_1 + X_0 Q_1$$

$$K_0 = \neg Q_1 + X_0 + X_1$$

$$Z = X_1 X_0 / Q_1 Q_0 + X_1 X_0 Q_1 / Q_0$$

c) Realizar el diagrama de estados utilizando los estados necesarios del diagrama adjunto, añadiendo más si fuera necesario o dejando en blanco los que no se necesiten.



d) Indicar la función que realiza la máquina de estados

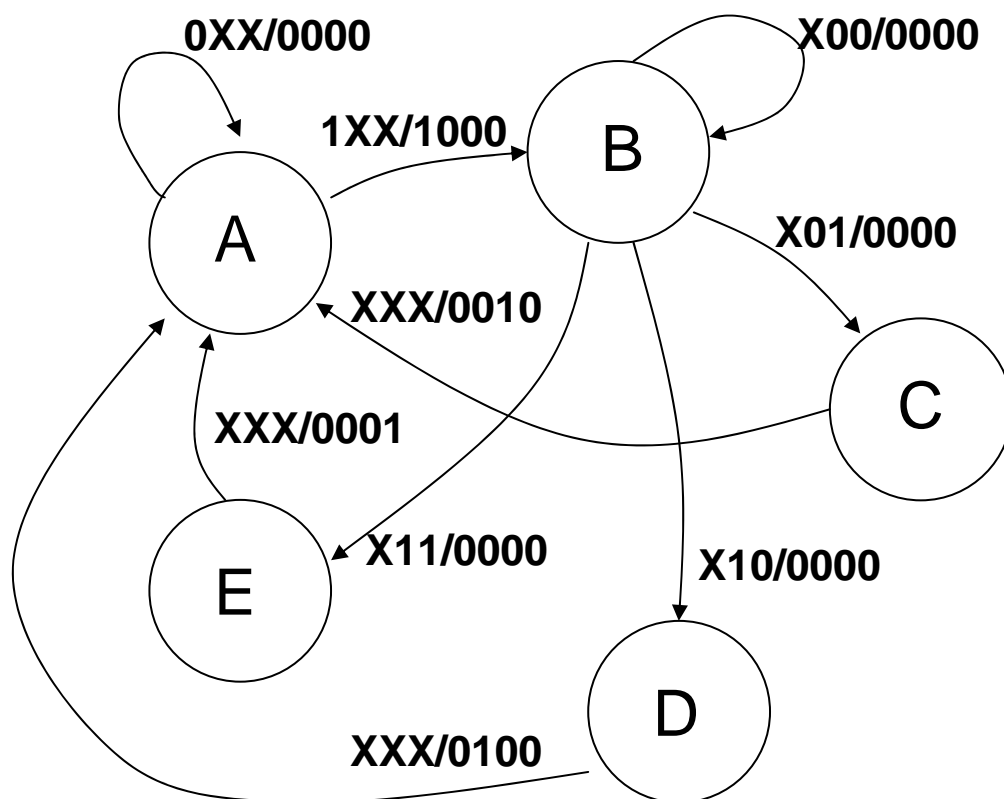
La máquina de estados detecta la secuencia 0010 sin solapamiento si $X_1=1$ y la secuencia sin solapamiento 0011 si $X_1=0$

P3.- Realizar una máquina de Mealy que controle el funcionamiento de una máquina expendedora de helados. Las entradas se denotarán con X_i y las salidas con Z_i . La máquina, tras introducir una moneda del valor adecuado (X_1) expulsa un cucurucho (Z_1). Tras ello da la posibilidad al usuario de elegir un helado de fresa (Z_2), de chocolate (Z_3) o de fresa y chocolate (Z_4). Para ello el cliente debe pulsar el botón correspondiente a “sabor fresa” (X_2), “sabor chocolate” (X_3), o ambos botones si desea el helado de sabor mixto. Una vez elegido el sabor, y tras un ciclo de reloj de espera (retardo) para dar tiempo al usuario a prepararse, se obtiene el helado colocando el cucurucho bajo el dispensador.

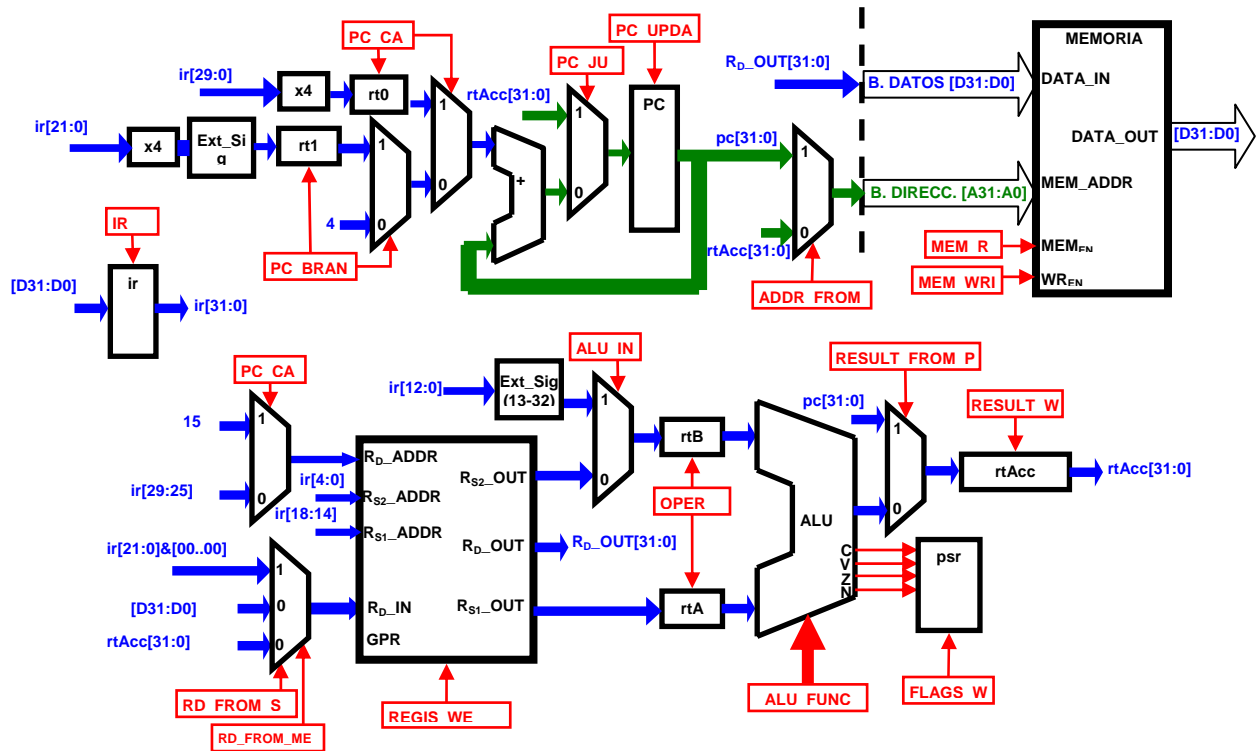
En el mismo ciclo de reloj que se dispensa el helado, el sistema vuelve al estado inicial y queda a la espera de un nuevo cliente.

Se supone que el cliente introduce siempre la moneda correcta, por lo que sólo controlaremos la presencia o ausencia de moneda, no el valor de la misma.

La nomenclatura a seguir es: $X_1X_2X_3 / Z_1Z_2Z_3Z_4$. Realizar el diagrama de estados utilizando los estados necesarios del diagrama adjunto, añadiendo más si fuera necesario o dejando en blanco los que no se necesiten.



P4.- En la figura se muestra la arquitectura del procesador ARC estudiado en teoría. Con los datos facilitados en la figura se pide:



Utilizando el lenguaje de transferencia de registros (RTL), explicar los cuatro ciclos de la ruta de datos necesaria para la ejecución de la instrucción **addcc**, señalando las distintas opciones de direccionamiento.

SOLUCION

- ① $[ir] \leq | OP \quad | RD \quad | OP1 \quad | RS1 \quad | \text{Inm13} \quad | \leq \text{MEM}[pc]$
 $[ir] \leq | OP \quad | RD \quad | OP1 \quad | RS1 \quad | \quad \quad | RS2 \quad | \leq \text{MEM}[pc]$
- ② IF $(ir[13]=1)$ THEN $[rtA] \leq [RS1]$; $[rtB] \leq \text{ext_sig}(\text{Inm13})$
ELSE $[rtA] \leq [RS1]$; $[rtB] \leq [RS2]$
- ③ $[rtAcc] \leq [rtA] + \text{---} [rtB]$
IF $(ir[24]=0)$ THEN $[psr] \leq (N,Z,V,C)$
- ④ IF $(ir[24:19] = 111000)$ THEN $[pc] \leq [rtAcc]$
ELSE $[RD] \leq [rtAcc]$; $[pc] \leq [pc] + 4$

P5.- Dado el siguiente programa escrito para ARC, en donde se mezclan instrucciones en ensamblador y en código máquina (hexadecimal). Se pide, **a)** utilizando la tabla adjunta, completar el código ensamblador. **b)** Indicar la función que ejecuta el código, señalando explícitamente el significado de la constante **start** y de la variable **long**.

op	Tipo	op2	Instrucción	[rd] = cc	Instrucción	op3	Instrucción
00	sethi/branch	010	branch	00001	be	000000	ld
01	call	100	sethi	00101	bcs	000100	st
10	ALU			00110	bneg	010000	addcc
11	Memoria			00111	bvs	010001	andcc
				01000	ba	010010	orcc
						010110	orncc
						100110	srl
						111000	jmp

PROBLEMA:	SOLUCION:
.begin	.begin
.org 2048	.org 2048
start .equ 4096	start .equ 4096
ld [long], %r1	ld [long], %r1
C4002830	ld [direcc], %r2
andcc %r3, %r0, %r3	andcc %r3, %r0, %r3
loop: 80884001	loop: andcc %r1, %r1, %r0
be done	be done
82807FFC	addcc %r1, -4, %r1
addcc %r1, %r2, %r4	addcc %r1, %r2, %r4
ld %r4, %r5	ld %r4, %r5
10BFFFFB	ba loop
addcc %r15, 4, %r15	addcc %r15, 4, %r15
done: jmp %r15 + 4, %r0	done: jmp %r15 + 4, %r0
long: 20	long: 20
direcc: start	direcc: start
.org start	.org start
a: 25	a: 25
-10	-10
33	33
-5	-5
7	7
.end	.end

b) Es una subrutina, finaliza con jmp, que lee datos de una tabla que se inicia en la posición de memoria señalada por una constante (start) y ocupa un determinado número de bytes dado por otra variable (long)